UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/651,425	08/30/2000	Christopher Songer	003048.P008	2566

23363	7590	09/29/2006

CHRISTIE, PARKER & HALE, LLP
PO BOX 7068
PASADENA, CA 91109-7068

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 09/29/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	09/651,425	SONGER ET AL.
	Examiner	**Art Unit**	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>30 June 2006</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-44* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date <u>See Continuation Sheet</u>.

4)☒ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. <u>9/25/06</u>.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :5/28/2004 (re-transmission w/ complete documents consideration by Examiner).

DETAILED ACTION

1.   This action is responsive to the Applicant's response filed 6/30/2006

As indicated in Applicant's submission, claims 4, 7, 11, 17, 22-24, 28, 32, and 39 have

been amended; and claims 1-44 are pending in the office action.

Specification

2.   The disclosure is objected to because of the following informalities: The term 'variant'

and 'varient' are referred to and used interchangeably in different parts of the Specifications and

the term 'varients' (e.g. pg. 27, lines 17-20) appears to be a misnomer, a misspell; and is to be

corrected to impart consistency for this reference.

3.   Further, the use of 'variant performs a function whose inputs and outputs are identical

outside of the function' (pg. 18, line 13-14) needs to be readjusted in light of an earlier

description giving a physical/configuration nature to this term called *variant* (Specifications:

design, bin, separation of memory, geometric layouts - pg. 18, lines 5-9). Correction to this

phrase is needed in order to maintain consistency to the above description, i.e. this variant being

merely a data implementing some design type of differential concerning a memory bin and/or its

internal configuration; as opposed to a function under execution (being performed) based on

inputs and providing outside world outputs.

Appropriate correction is required.

Claim Objections

4.   Claims 1, 22, 43-44 are objected to because of the following informalities: the limitation

recited as 'code variant performs a function whose inputs and outputs are identical' (last line of

respective claims). There is not sufficient teaching in the Specifications to support the usage of

the terminology such as 'code variant ... performs a function whose inputs and outputs are ...';

because the Specifications discloses not a code function --with inputs and outputs -- that is being

performed (see Specifications, pg. 18, lines 5-9; pg. 27, lines 4-7). The *variants* are rather

disclosed as configuration designs being implemented for a bin or memory block that would not

be visible to the outside world in terms of inputs and outputs to a memory section; and may

include differentials about layout within a bin; all of which not amounting to a particular

function being performed in the context of any execution requiring inputs into and yielding

outputs from such function. The language about a *code variant* performing a function with

inputs and outputs is to be readjusted to reflect the role of this disclosed limitation, which is

purely statically informative or configurative, not actively functional as a block of code under

execution.

At best, this *variant* limitation will be treated as a differing piece of configuration data set

relative to other sets and acting as virtually transparent to the outside world.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

6.   Claims 1, 3-22, 24-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over, in

view of Edwards et al., "Hardware/software partitioning for performance enhancement", 1995,

Partitioning in Hardware-Software Codesigns, IEE Colloquium, (hereinafter Edwards); in view

Mirsky et al., USPN: 5,915,123 (hereinafter Mirsky).

As per claim 1, Edwards discloses a method of creating run time executable code, the

execution using a processing element array (FPGA - pg. 1), comprising:

partitioning a processing element array into a plurality of hardware accelerators (e.g.

accelerated ..in hardware, 2nd para, pg. 1; Fig. 2 – *hardware ... point accelerator* – 3rd para, pg.

2; *partitioning into hardware* - 4th para, pg. 2);

identifying a plurality of functions in the program that are anticipated to consume a

substantial execution time (e.g. *hot spots* – 2nd para, pg. 2);

decomposing a program source code into a plurality of kernel sections to represent the

plurality of functions (e.g. *system partitioner, critical regions* – 5th para, pg. 2);

mapping said plurality of kernel sections into a plurality of hardware dependent

executable code for execution on the plurality of hardware accelerators(e.g. *hardware/software*

interface, HardwareC – Fig. 2; *placing and routing... array C ... adapted C* – 2nd para, pg. 3).

But Edwards does not explicitly disclose that the mapping is a matrix nor does Edwards

teach forming a matrix describing different combinations of said hardware accelerators, code

variants and said hardware dependent executable code entities configured to support run time

execution of the kernel sections by the processing element array wherein each variant performs a

function whose inputs and outputs are identical.

Edwards discloses execution of kernel sections by processing elements (Fig. 2; *placing*

and routing... array C ... adapted C – 2nd para, pg. 3) and set up of parameters and registers for

such hardware/software mapping for the improved execution of the plurality of the array element

(e.g. Fig. 3, pg. 4), the combinations of parameters and changes to registers thus reading on

variants being applied to the hardware/software partitioning and array execution. Mirsky, in a

system to partition a FPGA-like (MCPE) system of processing elements analogous to Edwards,

discloses code execution contexts and hardware mapping (Fig. 8, Fig. 15) according to which

MCPE can be grouped with flexible data/control configuration via use of a FSM controller for

controlling the MCPE interaction between the groups via a set of variants similar to the

parameters by Edwards, thus variants for identifying or selecting the MCPE (see e.g. masked

identification -col. 11 to col. 13; Fig. 19-23) or for reallocating memory areas, i.e. the variants

thus imparted appearing as virtually unchanging -- or whose input/outputs are identically

perceived from outside the contexts being thus partitioned; i.e. Mirsky allocate per group of

MCPE a context or group thereof along with memory for such PEs as well as the control data,

memory configuration and identification of transmission among the group thus partitioned (see

col. 14 li. 20 to col. 15, li. 20; Fig. 20-23). In view of the array disposition and the context

organization as from Fig. 8 by Mirsky, it would have been obvious for one skill in the art at the

time the invention was made to allocate parameters for executing the hardware-implemented

processing PE array by Edwards so that these are configured as a table of configuration data or

variants as taught by Mirsk in the context grouping from above, such table having dimensions

mapping memory contexts, MCPE and configuration data to support the runtime process of

accelerating the hot spots by Edwards. One would be motivated to do so because this

configuration table (or matrix) putting in evidence the dependency of control/configuration data,

memory context, and hardware processing elements would enable alleviate static resources

dependency by providing dynamic/more flexible re-accommodation of resources at runtime

according to architecture, distribution/storage and architecture of hardware or physical resources

(see Mirsky, BACKGROUND).

As per claim 3, Edwards teaches critical regions to be mapped into hardware accelerators

configuration language and implementation thereof (see Fig. 2) but does not teach partition of

bins. The regions being implemented by HDL or HardwareC by Edwards are enhanced by

Mirsky via context as set forth in claim 1 above, hence the combination Edwards/Mirsk has

disclosed partitioning into bins by virtue of claim 1.

As per claim 4, Edwards/Mirsky further discloses mapping includes mapping into

multiple hardware contexts (e.g. see Mirsky: Fig. 8).

As per claims 5 and 6, Edwards teaches parameters (see *configuration data*, 2nd para,

pg. 3; *register, parameter memory* – pg. 4) and Mirsky further discloses mapping (re claim 5) a

first set of variants to select region/address of activated PE (e.g. Fig.17, 19, 20-23 – Note:

context being chosen reads on variants based on resource usage). The rationale for obviousness

has been set forth in claim 1.

As per claim 7, Mirsky further discloses mapping a second set of variants configured to

support multiple hardware configurations of one of a plurality of bins (e.g. col. 20, lines 20-39;

Fig. 9, 15 – Note: context 0 ... context 3 reads on one bin with multiple hardware configurations

). The rationale for obviousness has been set forth in claim 1.

As per claim 8, Edwards discloses mapping is performed by a place and route (e.g.

placing and routing – 2nd para, pg 3).

As per claims 9 and 10, Edwards further discloses the decomposition step is performed manually (e.g. *static analysis, researchers use of...*) and software profiler (*our profiling* – 4th para, pg. 2; *performance profiler* – top para, pg.2).

As per claim 11, Edwards discloses monitoring timing of execution of code compiled from source code(e.g. *hot spots*, 2nd para -pg. 2).

As per claims 12 and 13, Edwards discloses utilizing set of test data (e.g. representative data – 4th para, pg. 2); determining functions that consume a significant portion of execution timing (e.g. *time stamps, determine where the program spends its time* – 2nd para, pg 2)

As per claim 14, Edwards discloses identifying functions by identifying regular structures (e.g. *C types, integers, unions* – 5th para, pg. 2).

As per claims 15 and 16, Edwards discloses identifying kernel sections by identifying C functions with a limited number of inputs and outputs (e.g. *C functions* – Fig. 2); or with a limited number of branches (e.g. Fig. 2 - Note: a skill in the art would view or a C functions or any basic blocks thereof as those having very limited number of branching)

As per claim 17, Edwards discloses decomposing by identifying overhead sections (e.g. C functions, Fig. 2; 2nd para, pg. 3 – Note: any C code when compiled would have to be segregated into header parts and body parts, the header setting the macro definition to the body of the main code).

As per claim 18, Edwards does not disclose microcode. Mirsky discloses the use of microcode (e.g. col. 10, lines 21-32). In view of the use resource-constrained processing elements such as chips of FPGA on-chip storage (see Mirsky: col. 1, line 40 to col. 2, line 23), it would have been obvious for one of ordinary skill in the art at the time the invention was made

to implement microcode to the processing element such as taught by Mirsky to Edwards' FPGA

processing elements. One of ordinary skill in the art would be motivated to do so because this

would alleviate memory storage in small device used as PEs in the array of Edwards' system,

and further improve resource usage in addressing the storage issue (as mentioned by Mirsky) of

such processing unit while trying to get code to accelerate critical areas of execution in

hardware/software codesign.

As per claim 19, Mirsky discloses that mapping includes creating context dependent

configurations (e.g. Fig. 15; Fig. 18-23) to enhance the i/o control implementation by Edwards'

approach in interacting the FPGA units(see Fig. 2, 3, bottom pg. 3, top pg. 4). The rationale as

to generating context associated with variants has been set forth in claim 1.

As per claims 20 and 21, Edwards does not explicitly teach that the matrix used in the

mapping is sparely (re claim 20) or fully (re claim 21) populated; but discloses the connectivity

or mapping of FPGA elements with respect to the testing parameters or configuration data (e.g.

Fig. 2, 3, bottom pg. 3, top pg. 4). The rationale using Mirsky' s approach using contexts and

variants being set forth in claim 1 to render the use of table mapping variants, code and hardware

accelerators into a matrix obvious would have render herein the limitations as to the density of

such matrix. That is, the limitation on such matrix being populated as claimed herein would

have been obvious because of the same rationale mentioned in claim 1; and also because

Edwards's configuration analysis as mentioned above would imply sparsely or fully populating

of the matrix as mentioned in claim 1, dependent of the nature/number of critical regions

identified (see Fig. 1, profiler) as well as the resources of the hardware at disposition, or variants

thereof.

As per claim 22, this claim is a system claim corresponding to claim 1 above and includes most of the limitations therein using Edwards's disclosure, namely system for runtime code execution on a processing element array, comprising:

a plurality of hardware accelerators partitioned (e.g. e.g. *accelerated ..in hardware*, 2nd para, pg. 1; Fig. 2 – *hardware ... point accelerator* – 3rd para, pg. 2; partitioning into hardware - 4th para, pg. 2);

plurality of kernel sections (e.g. *system partitioner, critical regions* – 5th para, pg. 2) anticipated to consume substantial execution time of a program source code, when executed on said accelerator (*hot spots* – 2nd para, pg. 2);

plurality of hardware dependent executable derived from said kernel sections for execution on said accelerators (e.g. . *system partitioner, critical regions* – 5th para, pg. 2);

a mapping of combinations of said hardware accelerators and kernel designs for run time execution (e.g. *hardware/software interface, HardwareC* – Fig. 2; *placing and routing... array C ... adapted C* – 2nd para, pg. 3).

But, like in claim 1, Edwards does not explicitly disclose that the mapping is a matrix nor does Edwards teach forming a matrix describing different combinations of said hardware accelerators, code variants and said hardware dependent executable code entities configured to support run time execution of the kernel sections by the processing element array wherein each variant performs a function whose inputs and outputs are identical.

However, these limitations have been addressed in claim 1.

As per claims 24-30, these are system claims corresponding to claims 3-9, respectively; hence, are rejected using the corresponding rejections set forth therein, respectively.

As per claims 31-38, these claims are system claims corresponding to claims 10-17, respectively, hence, are rejected herein using the corresponding rejections set forth therein, respectively.

As per claim 39, this claim corresponds to the hardware dependent executable including microcode of claim 18 above, hence, is rejected herein using claim 18 rejection as set forth above.

As per claim 40, this claim corresponds to claim 19 above, hence, is rejected herein using claim 18 rejection as set forth above

As per claims 41-42, these claims are similar to claims 20-21 above, respectively; hence are rejected herein using the same grounds set forth therein.

As per claim 43, this claim is a computer-readable medium version of claim 1, above hence includes all the step limitations therein and is rejected herein using the same corresponding rejections set forth therein.

As per claim 44, this claim is a system version of claim 1, above hence includes all the step limitations therein and is rejected herein using the same corresponding rejections set forth therein

7.   Claims 2 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Edwards et al., "Hardware/software partitioning for performance enhancement", 1995, and Mirsky et al., USPN: 5,915,123; as applied to claims 1 and 22, and further in view of Tseng et al., USPN: 6,009,256 (hereinafter Tseng).

As per claim 2, Edwards does not teach about partition of DSP. In a co-simulation using HW/SW analogous to the HW/S-based FPGA co-design by Edwards, Tseng discloses

partitioning into digital signal processors (*EAB, DSP* -- col. 51, lines 27-34). Since the

accelerators are used to accelerate execution of critical regions, it would have been obvious for

one skill in the art at the time the invention was made to implement the accelerator-designated

code execution by Edwards so that the hardware being used to accelerate the target runtime of

the designed integrated system are a network of processing elements as DSPs; because DSP are

known to have their own processor for supporting complex functions via latest/advanced DSP

architecture, thus enhancing the acceleration as intended by Tseng (see Hardware Acceleration-

col. 3, col. 8-9).

As per claim 23, this claim corresponds to claim 2 above, hence, is rejected herein using

the rejection as set forth therein.

Response to Arguments

8.   Applicant's arguments filed 6/30/06 have been fully considered but they not persuasive.

Following are the observations by the Examiner in regard thereto.

USC §103(a) Rejection:

(A)   Applicants have submitted that Edwards's FPGA being analogized to the MCPEs

standing for the recited hardware accelerators of the invention is considered incorrect because of

the how FPGA is defined relative to how the present Invention representing the MCPE

architecture (Appl. Rmrks, pg. 9, middle to pg. 10, 2nd para). It is noted that the term 'hardware

accelerators' is not recited in sufficient terms for the specific material proffered by Applicants

(Appl. Rmrks, pg. 9) to otherwise support the rationale as to why Edwards' FPGA would be

inapposite with the MCPEs of the disclosure. For one skill in the art, when array of elements are

partitioned so to support accelerated execution of code via implemented hardware as

contemplated by Edwards, with the implementation of such optimized code being shown as a

form of Hardware program or synthetizer, or FPGA with optimized stream synthesis, the notion

of hardware accelerators have been considered disclosed via such implementation; and this is

supported by the final execution using a HW System as presented in Figure 3 of Edwards. The

language of the claim as interpreted has enabled the mapping as set forth in the rejection. That

is, based on a software optimization stage leading to a hardware-implemented stage so that when

hardware devices are being mapped from the result of the Hardware language program as

purported by Figure 2 of Edwards, the resulting process being thus implemented via FPGA reads

on hardware accelerators being the result of the source code partitioning. There is no definition

of the 'accelerator' in the claim -- nor is there any explicit reciting that the accelerators thus

identified via partition are no longer part of a software partitioning tool but actually hardware

components tangibly destined for an actual HW runtime execution -- for this claimed subject

matter to preclude the implementation of HardwareC in conjunction with Figure 3 FPGA

synthesizer by Edwards from reading on or being analogized to what is construed from the term

'hardware accelerator'. As set forth in the rejection, the very implementing using hardware

constructs (Fig. 2 – *hardware ... point accelerator* – 3rd para, pg. 2; *partitioning into hardware* -

4th para, pg. 2) --via HW synthesizer or HardwareC-- from an earlier improved source code thus

partitioned reads on the recited limitation (*). The steps recited as partitioning, identifying,

decomposing, mapping and forming a matrix are more reasonably understood as being done in a

software/HW simulation/mapping founded on code optimization/analysis stage; which as a

whole is commensurate to Edwards' tool and process steps. Applicant's arguments fail to

comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define

a patentable invention without specifically pointing out how the language of the claims

patentably distinguishes them from the references. At one time, an accelerator is disclosed a just

a bin (see SUMMARY of invention) within an array of MCPE; and this is not the same as

hardware elements as proffered as Fig. 2 in the Remarks (pg. 9), making it very evident that

deliberate and unique definition of the 'hardware accelerator' has not been provided. The

Specifications will not be read into the claims; therefore, it is construed that any component

making up the FPGA (i.e. a processing element array) can be considered a hardware accelerator (

refer to *), absent any teaching anywhere in the claim that hardware accelerator will be

independent to or not integral to a FPGA. The argument about distinguishing difference between

electrically interconnect points and processing element of Figure 2 (e.g. *hardware accelerator*

claimed instead on processing elements being a ALU) would not be commensurate with the

language of this claimed limitation; and the argument is not persuasive.

(B)   Applicants have submitted that translation by Edwards of C code in HardwareC and

producing FPGA does not constitute 'partitioning the processing element array into a plurality of

... accelerators' (Appl. Rmrks, pg. 10, 3rd para). The argument will be referred to section A

above, notably because there is not sufficient defining of the 'accelerators' concept from the

disclosure; nor is there any description in the claim that would prevent the hardware constructs (

see Edwards: *synthesizer*, and Fig. 3) implementing the HardwareC program to read on

accelerators in light of Edwards' optimization and partitioning of source code. The argument

about Edwards not disclosing 'mapping ... kernel sections ... executable code' has not set forth

how the cited parts of the rejection have failed to disclose this.

(C)　Applicants have submitted that there is not motivation to combine processing elements array by Mirsky to the C code translation of Edwards because of the fundamental difference between FPGA and PE array, making the two references incompatible with each other (Appl. Rmrks, pg. 11, 1st para). It is noted that FPGA is but a global approach to simulating or observing streams of interconnected processing elements, and a PE can be considered one of the components comprising this array. The claimed 'mapping ... into a plurality of hardware dependent executable code for execution on the plurality of hardware accelerators' does not enforce a runtime execution of an ALU or a Block of memory controller, or any PE in a hardware runtime per se. When the mapping process is interpreted from the claim, this process reads on an analysis stage in order for particular code sections to be used on a corresponding eventual hardware component processing such code sections instructions, such component being part of the FPGA in conjunction with the synthesizer supporting Hardware program constructs as taught by Edwards. And the rejection has cited portions that fulfill such interpretation. Based on Edwards teaching, the rationale to combine using Mirsky does not fall into what is perceived a non-analogous teaching as alleged from above.

(D)　The rest of the arguments (Appl. Rmrks, pg. 11, 12) are based on Edwards' failure to disclose what Applicants believe to be the processing element array as required by claim 1; and these arguments are also not persuasive by virtue of the response in section A.

The claims will stand rejected as set forth in the Office Action.

Information Disclosure Statement

9.   Applicants are now provided with a duplicate copy of PTO-1449 regarding an earlier IDS

filed as of 5/28/04; and according to which all documents have now been considered and this

should solve any regrettable confusion that this IDS resubmission might have or had created.

Conclusion

10.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner

should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can

normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is

assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
September 25, 2006

KAKALI CHAKI
SUPERVISORY PATENT EX/
TECHNOLOGY CENTER